Lab 5: Rejection Sampling

Yutong Shao

Feb.6, 2023

Agenda

We can often end up with posterior distributions that we only know up to a normalizing constant. For example, in practice, we may derive

$$p(\theta \mid x) \propto p(x \mid \theta)p(\theta)$$

and find that the normalizing constant $p(x)$ is very difficult to evaluate. Such examples occur when we start building non-conjugate models in Bayesian statistics.

Given such a posterior, how can we appropriate it's density? One way is using rejection sampling. As an example, let's suppose our resulting posterior distribution is

$$f(x) \propto sin^2(\pi x), x \in [0, 1].$$

In order to understand how to approximate the density (normalized) of f, we will investigate the following tasks:

1. Plot the densities of $f(x)$ and the Unif(0,1) on the same plot. According to the rejection sampling approach sample from $f(x)$ using the Unif(0,1) pdf as an enveloping function.
2. Plot a histogram of the points that fall in the acceptance region. Do this for a simulation size of 10^2 and 10^5 and report your acceptance ratio. Compare the ratios and histograms.
3. Repeat Tasks 1 - 3 for Beta(2,2) as an enveloping function.
4. Provide the four histograms from Tasks 2 and 3 using the Uniform(0,1) and the Beta(2,2) enveloping proposals. Provide the acceptance ratios. Provide commentary.
5. (i) Do you recommend the Uniform or the Beta(2,2) as a better enveloping function (or are they about the same)? (ii) If you were to try and find an enveloping function that had a high acceptance ratio, which one would you try and why? For part (ii), either back up your choice using a small paragraph or (ii) empirically illustrate that it's better. (If you wanted to go above and beyond, you could do both, which is highly encouraged and would really prepare you for the exam!)

```
knitr::opts_chunk$set(echo = TRUE)
set.seed(2023)
```

Task 1

Plot the densities of $f(x)$ and the Unif(0,1) on the same plot.

Let's first create a sequence of points from 0 to 1, so that we can have a grid of points for plotting both of the proposed functions.

```r
# grid of points
x <- seq(0, 1, 10^-2)
```

```r
fx <- function(x) sin(pi * x)^2
plot(fx, xlim = c(0,1), ylim = c(0,1.5), ylab = "f(x)", lwd = 2)
curve(dunif, add = TRUE, col = "blue", lwd = 2)
curve(dbeta(x,2,2), add = TRUE, col = "red", lwd = 2)
legend("bottom", legend = c(expression(paste("sin(",pi,"x)"^"2")),"Unif(0,1)",
"Beta(2,2)"), col = c("black", "blue", "red"), lty = c(1,1,1), bty = "n", cex = 1.1, lwd = 2)
```

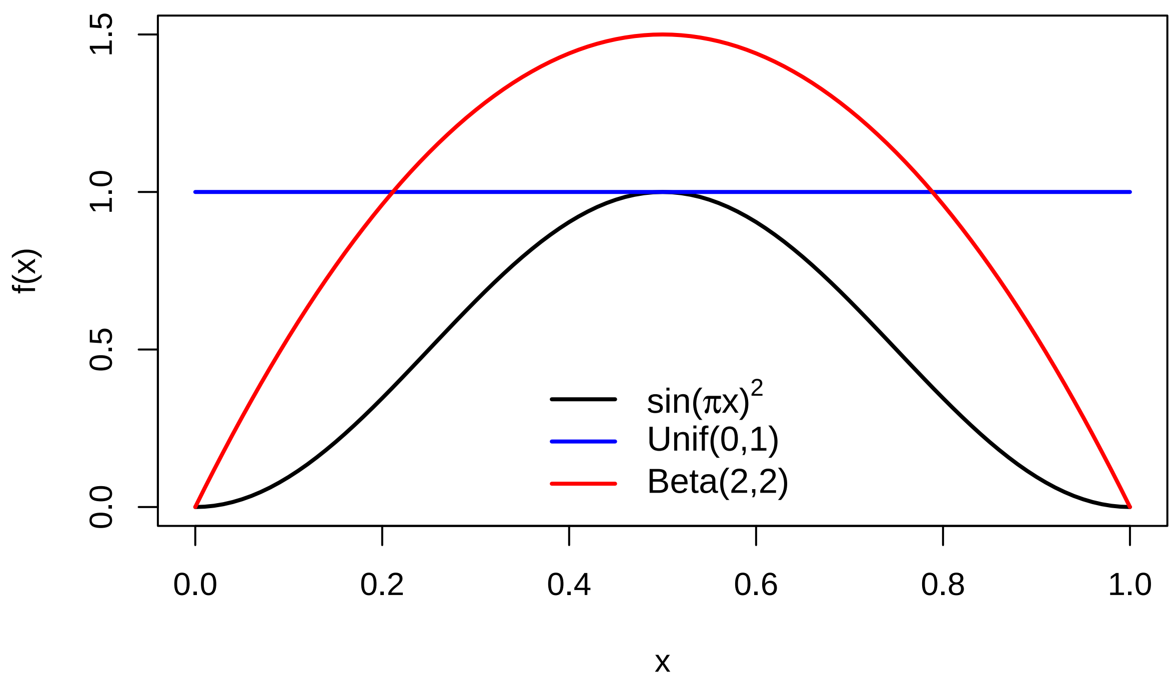


Figure 1: Comparision of the target function and the Unif(0,1) and the Beta(2,2) densities on the same plot.

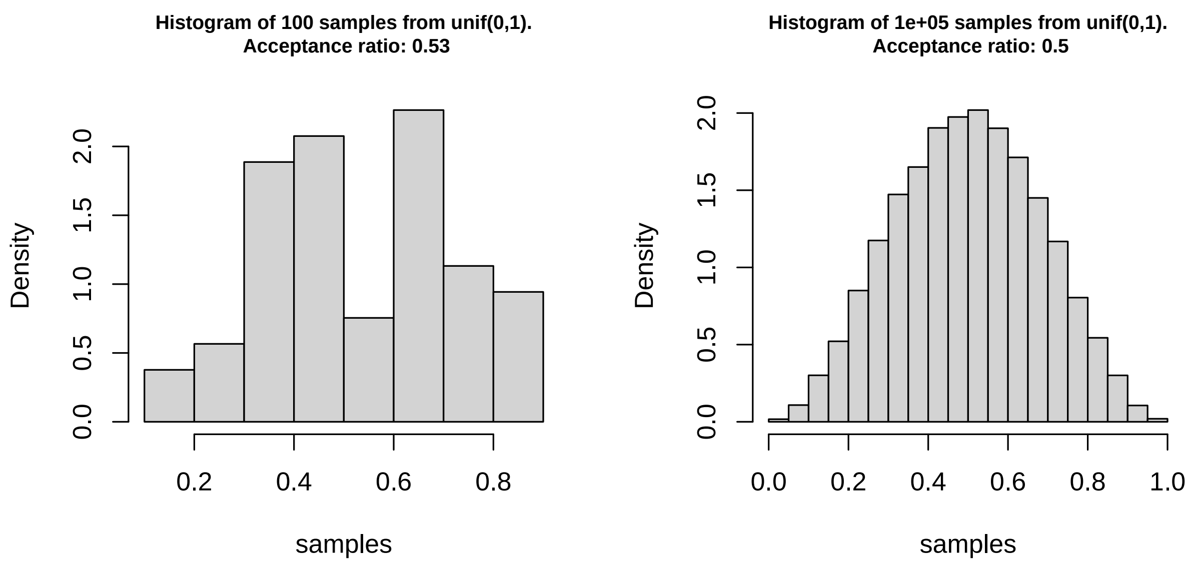
Tasks 2

According to the rejection sampling approach sample from f(x) using the Unif(0,1) pdf as an enveloping function. In order to do this, we write a general rejection sampling function that also allows us to plot the historams for any simulation size. Finally, our function also allows us to look at task 4 quite easily.

```r
sim_fun <- function(f, envelope = "unif", par1 = 0, par2 = 1, n = 10^2, plot = TRUE){
  # f: target func
  r_envelope <- match.fun(paste0("r", envelope))
  d_envelope <- match.fun(paste0("d", envelope))
  proposal <- r_envelope(n, par1, par2)
  density_ratio <- f(proposal) / d_envelope(proposal, par1, par2)
```

```
  samples <- proposal[runif(n) < density_ratio]
  acceptance_ratio <- length(samples) / n
  if (plot) {
    hist(samples, probability = TRUE,
         main = paste0("Histogram of ",
                       n, " samples from ",
                       envelope, "(", par1, ",", par2,
                       ").\n Acceptance ratio: ",
                       round(acceptance_ratio,2)),
                       cex.main = 0.75)
  }
  list(x = samples, acceptance_ratio = acceptance_ratio)
}
```

```
par(mfrow = c(1,2))
unif_1 <- sim_fun(fx, envelope = "unif", par1 = 0, par2 = 1, n = 10^2)
unif_2 <- sim_fun(fx, envelope = "unif", par1 = 0, par2 = 1, n = 10^5)
```




```
# ATTN: You will need to add in the Beta(2,2) densities on your own to finish task 4.
```

Figure 2: Comparision of the output of the rejection sampling for 100 versus 100,000 simulations with uniform and beta distributions as envelope functions.

```
par(mfrow = c(1,1))
```

The acceptance ratio is 0.48 and 0.5 for sample size of 10^2 and 10^5, respectively. Technically, they should all converge to 0.5, which is just the area under the target distribution (already been normalized) divided by the area under the envelope function (uniform distribution on [0,1]).

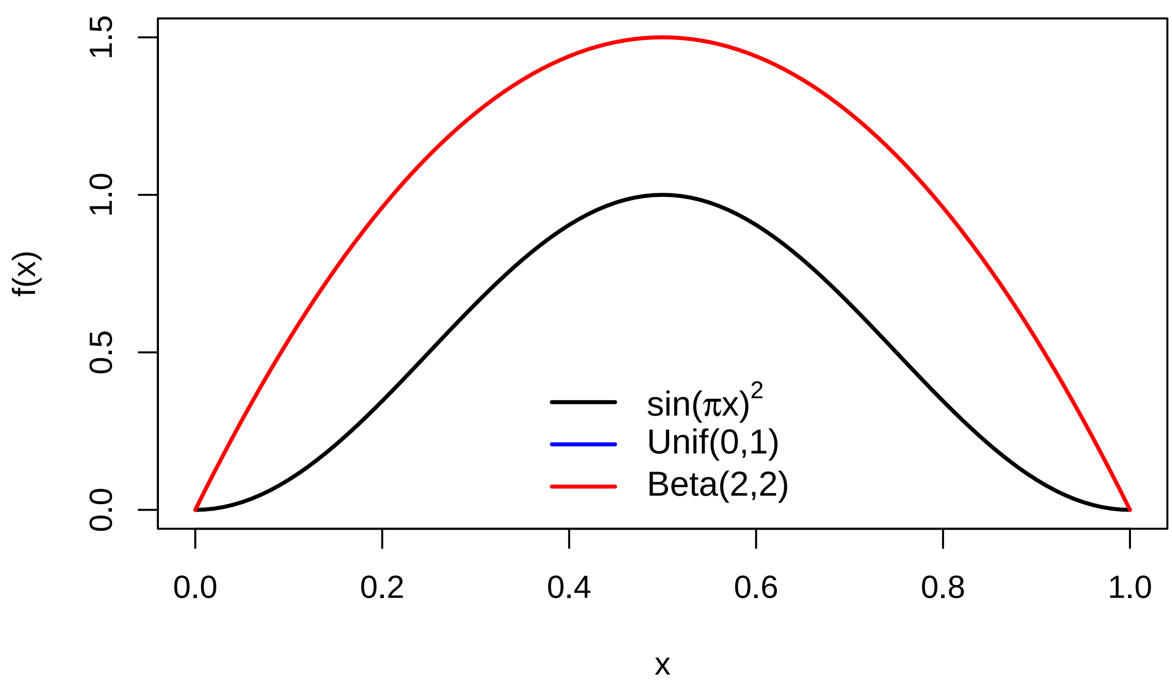
$$\text{Rejection rate} = \frac{\int_0^1 f(x)dx}{\int_0^1 U(x)dx} = \frac{1}{2}$$

And the histogram show the influence of sample size on rejection sampling. In general, rejection sampling works better with large sample size.

Task 3

Repeat Tasks 1 - 3 for Beta(2,2) as an enveloping function.

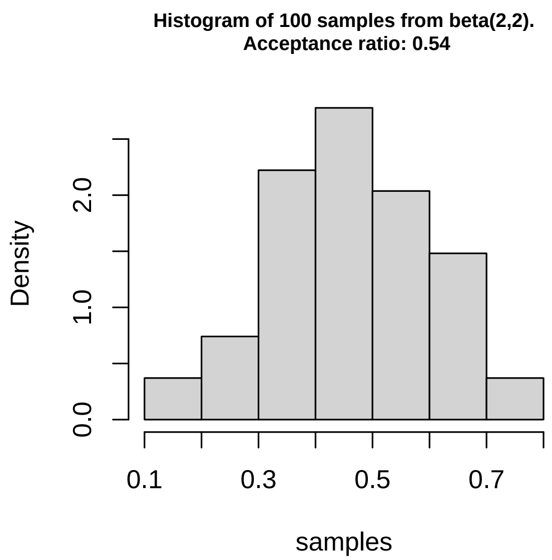
```
fx <- function(x) sin(pi * x)^2
plot(fx, xlim = c(0,1), ylim = c(0,1.5), ylab = "f(x)", lwd = 2)
# curve(dunif, add = TRUE, col = "blue", lwd = 2)
curve(dbeta(x,2,2), add = TRUE, col = "red", lwd = 2)
legend("bottom", legend = c(expression(paste("sin(",pi,"x)"^"2")),"Unif(0,1)",
"Beta(2,2)"), col = c("black", "blue", "red"), lty = c(1,1,1), bty = "n", cex = 1.1, lwd = 2)
```

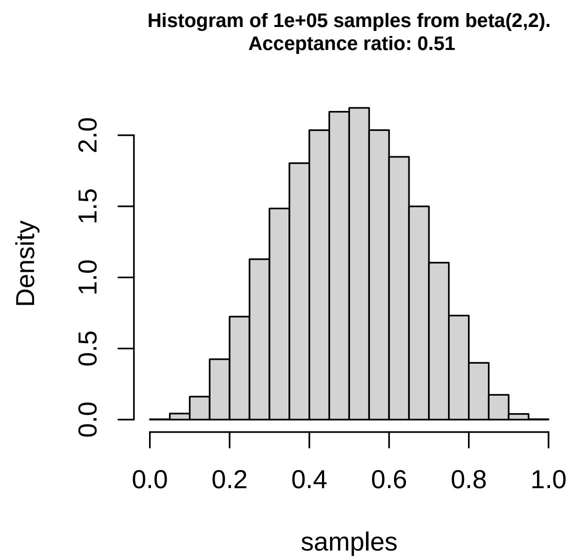



```
sim_fun_Beta <- function(f, envelope = "beta", a=2, b=2,
                         n = 10^2, plot = TRUE){
  # f: target func
  r_envelope <- match.fun(paste0("r", envelope))
  d_envelope <- match.fun(paste0("d", envelope))
  proposal <- r_envelope(n, a, b)
  density_ratio <- f(proposal) / d_envelope(proposal, a, b)
  samples <- proposal[rbeta(n, a, b) < density_ratio]
  acceptance_ratio <- length(samples) / n
  if (plot) {
```

```r
    hist(samples, probability = TRUE,
         main = paste0("Histogram of ",
                       n, " samples from ",
                       envelope, "(", a, ",", b,
                       ").\n Acceptance ratio: ",
                       round(acceptance_ratio,2)),
                       cex.main = 0.75)
  }
  list(x = samples, acceptance_ratio = acceptance_ratio)
}
```

```r
par(mfrow = c(1,2))
beta_1 <- sim_fun_Beta(fx, envelope = "beta", a=2, b=2, n = 10^2)
beta_2 <- sim_fun_Beta(fx, envelope = "beta", a=2, b=2, n = 10^5)
```





Task 4

Histograms are as above. Acceptance ratio:

```
## [1] "Sample size = 10^2"

## [1] "Acceptance ratio of Uniform(0,1) as envelope function is "

## [1] 0.53

## [1] "Acceptance ratio of Beta(2,2) as envelope function is "

## [1] 0.54

## [1] "Sample size = 10^5"
```

```
## [1] "Acceptance ratio of Uniform(0,1) as envelope function is"

## [1] 0.50098

## [1] "Acceptance ratio of Beta(2,2) as envelope function is "

## [1] 0.50757
```

For two kinds of envelope functions, the acceptance ratios are both close to 0.5 with large sample size. But the ratio of 0.5 is not a very ideal value since half the data points are wasted. When sample size is smaller, the acceptance ratio is a little bit higher due to small sample disturbance.

Task 5

1.

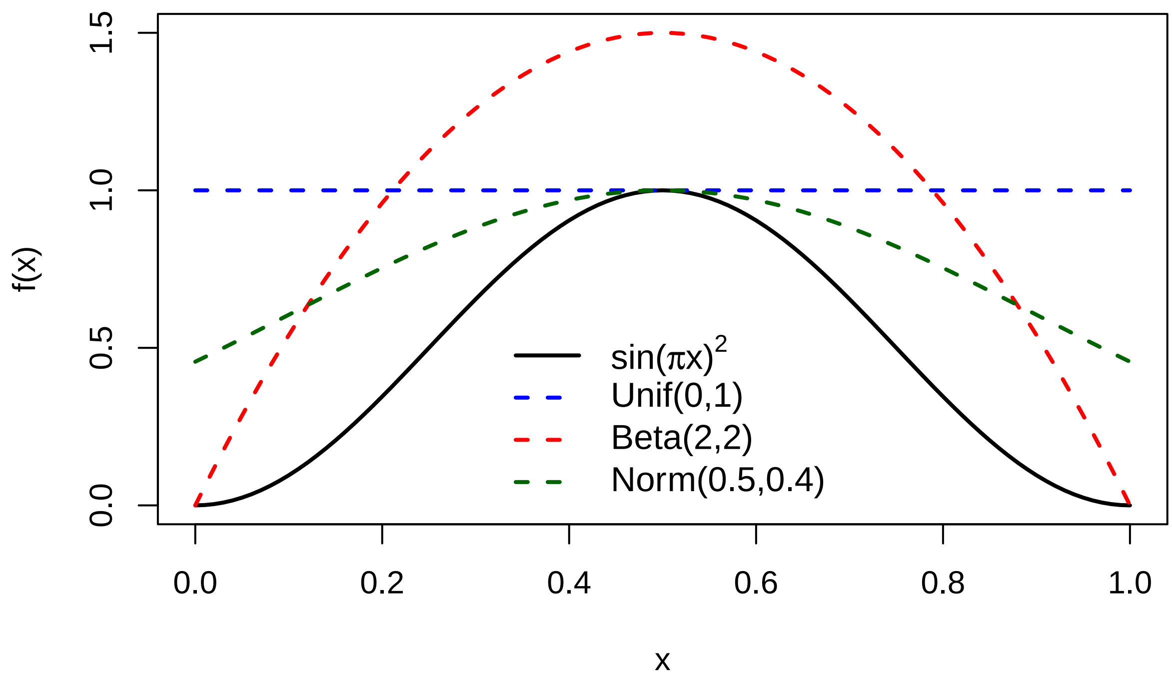
Since the acceptance ratio of both $Unif(0,1)$ and $Beta(2,2)$ are close to 0.5 when sample size is large, I think both functions yield similar result and are analogously effective.

2.

I think $N(0.5, \frac{1}{\sqrt{2\pi}}), (\frac{1}{\sqrt{2\pi}} \approx 0.4)$ is a better choice that can yield higher acceptance ratio.

I will prove my choice by first plotting my proposal function and target function. Previous examples are also included for reference.

```
fx <- function(x) sin(pi * x)^2
plot(fx, xlim = c(0,1), ylim = c(0,1.5), ylab = "f(x)", lty=1, lwd = 2)
curve(dunif, add = TRUE, col = "blue", lwd = 2, lty=2)
curve(dbeta(x,2,2), add = TRUE, col = "red", lwd = 2, lty=2)
curve(dnorm(x, mean = 0.5, sd=1/sqrt(2*pi)), add = TRUE, col = "darkgreen", lwd = 2, lty=2)
legend("bottom",
       legend = c(expression(paste("sin(",pi,"x)"^"2")),
                  "Unif(0,1)",
                  "Beta(2,2)",
                  "Norm(0.5,0.4)"),
       col = c("black", "blue", "red", "darkgreen"),
       lty = c(1,2,2,2), bty = "n", cex = 1.1, lwd = 2)
```

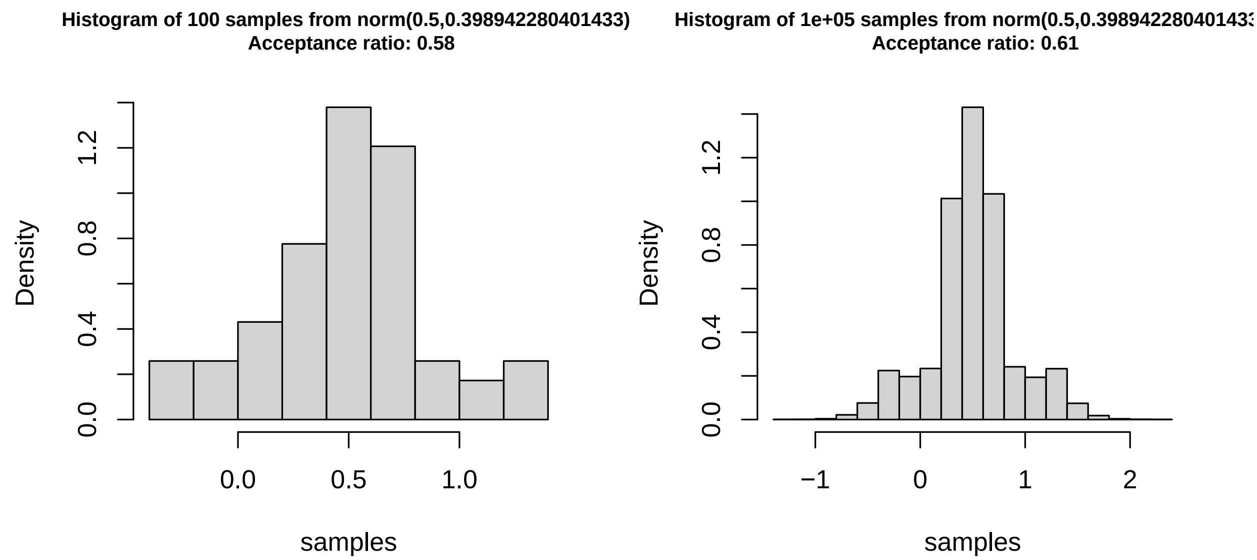


I repeat the above steps to prove the acceptance ratio of $N(0.5, \frac{1}{\sqrt{2\pi}})$ is higher than all the above proposal functions.

```
sim_fun_norm <- function(f, envelope = "norm", mu=0.5, sd=1/sqrt(2*pi),
                         n = 10^2, plot = TRUE){
  # f: target func
  r_envelope <- match.fun(paste0("r", envelope))
  d_envelope <- match.fun(paste0("d", envelope))
  proposal <- r_envelope(n, mu, sd)
  density_ratio <- f(proposal) / d_envelope(proposal, mu, sd)
  samples <- proposal[rnorm(n, mu, sd) < density_ratio]
  acceptance_ratio <- length(samples) / n
  if (plot) {
    hist(samples, probability = TRUE,
         main = paste0("Histogram of ",
                       n, " samples from ",
                       envelope, "(", mu, ",", sd,
                       ").\n Acceptance ratio: ",
                       round(acceptance_ratio,2)),
                       cex.main = 0.75)
  }
  list(x = samples, acceptance_ratio = acceptance_ratio)
}

par(mfrow = c(1,2))
norm_1 <- sim_fun_norm(fx, envelope = "norm", n = 10^2)
```

```r
norm_2 <- sim_fun_norm(fx, envelope = "norm", n = 10^5)
```



The acceptance ratio is 0.6 which is higher than 0.5.

First, normal distribution is a symmetric distribution, so it's a good candidate. Second, the mean should be 0.5. Third, the reasoning behind the value of standard deviation is setting the maximum value of normal distribution to 1 (which is the same as the maximum value of target function), and then compute σ.

$$\max_{\sigma} N(x \mid \mu = 0.5) = \max_{\sigma} \frac{1}{\sigma\sqrt{2\pi}} \exp\{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\} = 1$$

Apparently, the function reaches maximum when $x = 0.5$, thus,

$$\frac{1}{\sigma\sqrt{2\pi}} = 1 \Longrightarrow \sigma = \frac{1}{\sqrt{2\pi}}$$